



Mac OS X Networking Overview

Session 803





Mac OS X Networking Overview

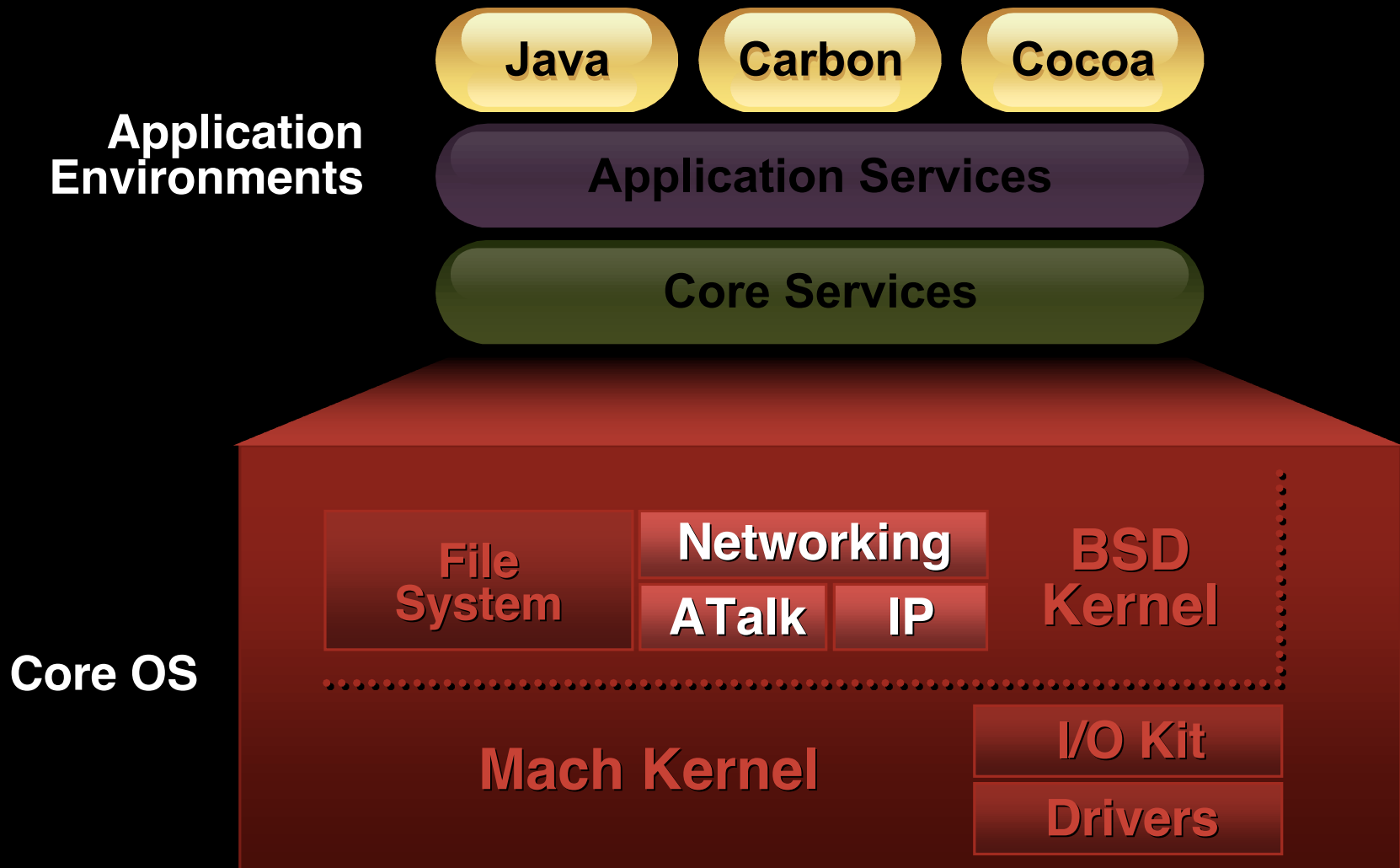
Vincent Lubet
Manager, Core OS Networking Team

Agenda

- Networking architecture
- What's new
- What are the APIs
- Hints and tips
- What's next



Networking in Mac OS X



Networking in Mac OS X

Ongoing Goals

- Ease of use
- Performance
- Better extensibility
- Standards compliance



Networking in Mac OS X

Architecture

- Extensible
- TCP/IP and AppleTalk protocol stacks
- Ethernet and PPP drivers
- IP Firewall and NAT
- Dynamic configuration
- ZeroConf



Core Networking

Based on FreeBSD

- Robust and proven implementation
- Popular API (Sockets)
- Easy-to-port Unix-like applications
 - Lots of open-source code available
- New in Jaguar
 - FreeBSD 4.4 network stack in Jaguar
 - Up-to-date IPFirewall and natd
 - IPv6 and IPSec based on KAME
 - PPTP



Networking in Mac OS X

Apple added value to FreeBSD

- Multi-threaded and MP efficient
- Tuned network buffer allocation
- Extensible—no need to recompile the kernel
 - Kernel development comes with lots of responsibility
 - More about NKE in Session 809



IPv6

- Available in Jaguar
- Addresses Internet growth concerns
- Automatic configuration
- APIs to be address family agnostic
- See Session 809 for more info



IPSec

- IETF standard to secure Internet traffic
- Jaguar has APIs and protocol implementation
- Foundation for VPN solutions
- For IPv4 and IPv6
- See Session 809 for more info



PPP

- Based on pppd
- Apple enhancements
 - PPPoE
 - CCL scripts
 - OT/PPP control API
 - Integrated in SystemConfig
- New on Jaguar PPTP client
 - Windows VPN connectivity



Also New in Jaguar

- A lot of bug fixes
- Performance improvements
- Detaching network interfaces
- New and updated man pages
- PF_NDRV works (userland protocol stacks)
- SNMP (net-snmp)



Network Configuration

- Preferences specify network “services”
 - Ethernet
 - AirPort
 - PPP
- More than one “service” can be active at a time
- Configuration dynamically updated
- Managed using System Configuration framework



System Configuration Framework

- APIs providing access to requested configuration
- APIs providing access to current state of the network
- APIs to check network accessibility
 - “Am I connected”
 - Replacement for TCPWillDial
- APIs to be notified when something changes
- Available in 10.1 and later
- MoreSCF DTS sample code



Rendezvous

- New in Jaguar
- Local networking that “always” work
- Easy to use like AppleTalk but . . .
- . . . using industry-standard TCP/IP
- Make existing network products better
- Make possible entirely new products
- See Session 811 for more info



Coding for Network Applications

Do not poll

- Polling uses 100% of CPU
- Hurts other processes and uses more power
- Instead block or be event driven
- Multiplex endpoints



Coding for Network Applications

Buffer sizes

- Size of buffers is critical for networking performance
 - Too small == high context switch overhead
 - Too large == starves VM for buffers
- Things to tune
 - Socket buffer size
 - Size of buffer with end/receive calls
- Receive socket buffer size directly affects TCP window size



Coding for Network Applications

Dynamic configuration

- Computer does not have a single IP address
- IP addresses change over time
- Servers should bind to the “Any IP” address
- Clients bind to nil
- More than one address is the norm: IPv4 and IPv6
- Do not assume IPv4 (SIOCGIFCONF)
 - Use `getifaddrs()` instead



Coding for Network Applications

Privileged operation

- Bind to low number port
 - Less than 1024
- Open raw socket
- Open bpf device
- Open PF_NDRV socket
- Open PF_KEY socket
- See DTS sample code AuthSample



Sockets API

- Native API to get the most out of the OS
- Lots of publicly available code
- *Unix Network Programming* by Stevens
- Carbon applications can eliminate sockets emulation layer over OT
- Mac OS X has a rich set of higher level networking APIs above Sockets





Networking APIs Above the OS

Becky Willrich
Application Frameworks

Networking APIs Above the OS

- What APIs are available
- Each API in depth
 - What it does
 - Strengths
 - Limitations
 - When should you use it
 - What's new



Networking APIs Above the OS

- URL Access (Carbon)
- Internet Config (Carbon)
- NSURL and NSURLHandle (Cocoa)
- NSNetService and NSNetServiceBrowser (Cocoa)
- CFNetwork (Core Services)
- WebServicesCore (Core Services)
- CFURL, CFSocket, CFStream (Core Services)
- Open Transport (Core Services)



How Does It Fit In?

Carbon

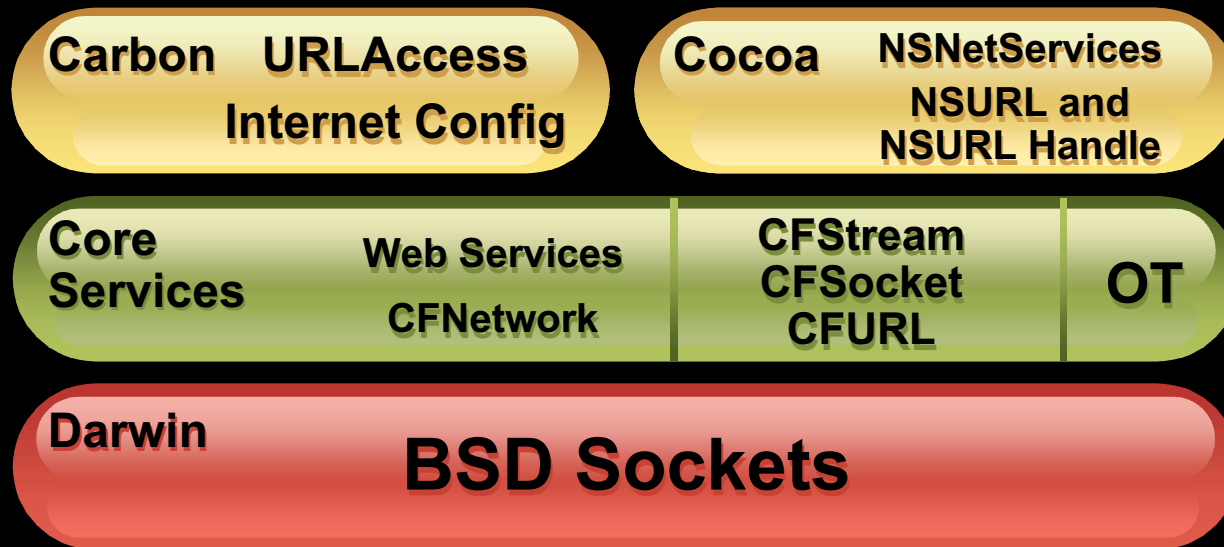
Cocoa

Core Services

Darwin



How Does It Fit In?



URL Access

- Provides APIs for up- and downloading URLs
- Focused on “just get the URL”
 - Automatically picks up system settings
 - Few scheme-specific options
- Supports http, https, ftp, file
- Part of Carbon; compatible back to Mac OS 8.6



URL Access—Strengths

- Simple, high-level API
- Options for extra processing on download
 - Populate a file/directory
- Options for adding UI elements (progress panel, authentication)
- Source-compatibility to Mac OS 9



URL Access—Limitations

- No way to add support for missing schemes
- API is showing its age
- Unimplemented features
 - Progress bar via `URLUpload()/URLDownload()`
 - Automatic decompression
- Some issues around pthreads and cooperative threads co-existing



URL Access—When to Use It

- If you need a strategy for both Mac OS 9 and Mac OS X
- If you need a scheme-agnostic, Carbon strategy
- If you don't want to get involved in the details of the download



URL Access—What's New

- Progress display from `URLOpen()`
- Unsuccessfully transferred files cleaned off the disk
- Redirection enabled
- Better event handling and notification
- Better error handling



Internet Config

- Holds “networking” settings
 - Web proxies
 - Helper apps for downloading kinds of URLs
 - MIME type to type/creator to extension mappings
 - App-specific preferences
 - Some user preferences (mail account, . . .)
- Legacy Carbon API



Internet Config—Strengths

- Well-known, well-understood API
- Only option on Mac OS 9
- Source compatibility with Mac OS 9



Internet Config—Limitations

- Implementation on Mac OS X is inefficient
 - Using ICBegin/ICEnd helps
- Some settings are dated or insufficient
- Some settings are completely obsolete
 - GetConfigFile and friends
- No longer the repository for many settings
 - Calls through to System Configuration or Launch Services in many cases
- Forces higher-level linkage than warranted



Internet Config—When to Use It

- Use IC only if there is no other choice
- Use System Configuration for proxy settings
- Use Launch Services to open particular files or URLs

LSOpenFSRef(), LSOpenCFURLRef()

- Launch Services also provides MIME type to handler mappings
- Use CFPreferences for app-specific preferences
 - This is a performance burden for all IC clients



Internet Config—What's New

- Nothing . . .



NSURL and NSURLHandle

- Cocoa “get the URL” API
- Two classes inside Foundation
 - NSURL represents a URL (bridged to CFURL)
 - NSURLHandle actually performs the up/download
- Primarily designed as an infrastructure for handling URLs
- http, https, ftp, file supported



NSURLHandle—Strengths

- Integrated with the run loop
 - Asynchronous APIs do not spawn threads
- Extensible via subclassing
- Well integrated into Cocoa APIs
 - [[NSImage alloc]
initWithContentsOfURL:myURL];**
- Good performance under load (used by Mail from the main thread for HTML mail)



NSURLHandle—Limitations

- Many “extras” are missing
 - No UI integration
 - No decompression or content handling
- Documentation is incomplete
- Many Cocoa APIs still refuse non-file URLs



NSURLHandle—When to Use It

- You want a simple download abstraction and URL Access isn't suitable
 - Available to Carbon X apps, too!
- You need to add your own scheme handler



NSURLHandle—What's New

- FTP support added
- Many bugs shaken out
- Asynchronous domain name lookup
- Integration with System Configuration planned for Jaguar



NSNetServices

- Allows you to register and discover services available on the Net
- Two classes
 - NSNetService
 - NSNetServiceBrowser
- New in Jaguar
- Implemented on CFNetServices in CFNetwork



CFNetwork

- Low-level abstractions for networking concepts
 - **NOT** a URL library
- APIs in the style of CoreFoundation
- Contains
 - SSL/TLS socket streams
 - A full HTTP engine
 - Net Services APIs
- Focus on high performance and small footprint
- Used by NSURLHandle, Sherlock, Software Update, iPhoto, WebServicesCore . . .



CFNetwork—Strengths

- Complete control of the HTTP transaction
 - Request can be manipulated directly, in HTTP-native terms
 - You control when bytes are read/written
- HTTP/1.1 fully supported
 - Pipelined, persistent connections
 - Digest authentication
- Full runloop integration
(schedule on multiple threads if you want)
- Best performance available above raw sockets



CFNetwork—Limitations

- Not a simple, convenient API
- No FTP support
- Little documentation



CFNetwork—When to Use It

- You know the transport is HTTP
- You want a simple, SSL-encrypted socket
- You want more control than the higher level APIs provide
- You want to avoid linking higher frameworks
- You need features only in CFNetwork



CFNetwork—What's New

- HTTP/1.1 persistent connections
- Net services and service discovery APIs
- SOCKS proxy support
- Digest authentication
- Asynchronous hostname lookup



WebServicesCore

- Client framework for web services
- New in Jaguar
- CFNetwork based
 - Exports CFTypes
 - RunLoop friendly
- Supports authentication options
- HTTP/S POST support only



Core Foundation

- Four interesting CFTypes
 - CFURL represents a URL string
 - CFSocket manages a socket on the run loop
 - CFReadStream and CFWriteStream are one-way byte streams to/from file, socket, or memory
- Also some dirt-simple functions for fetching URLs



Core Foundation—Strengths

- Single abstraction (CFStream) for a variety of input/outputs
- Strong run loop integration
- Supports a variety of threading models
- CFSocket supports arbitrary socket configurations
 - UDP as well as TCP
 - Server as well as client



Core Foundation—Limitations

- API is thin
 - Configure sockets via the BSD APIs
 - Need higher APIs for more complex operations
- No way to create new kinds of streams
- Little documentation



Core Foundation— When to Use It

- You want to manage a socket on the run loop
- You want to abstract away the actual source or destination of data
- You need to use an API that requires a `CFReadStream` or a `CFWriteStream`



Core Foundation— What's New

- CFReadStream/SetProperty()
- Write callbacks on CFSocket
- More options controlling your CFSocket callback
- Better event dispatching for CFStream
- Asynchronous hostname lookup
- Socket streams errored out on network configuration change
- SIGPIPE suppressed on socket streams



Open Transport

- Compatibility API for Mac OS 8 and 9
- Intended for easy migration to Mac OS X
- Emulation of Mac OS execution levels costly
 - Lots of threads and context switches
- No access to IPv6 and IPSec APIs
- New Mac OS X networking code should not use OT



IPv6

- CFSocket is IP-agnostic
 - You still must specify which version you want
- CFStream, CFNetwork, NSURLHandle, and URL Access all work with hostnames or URLs
 - We do the IPv6 work for you
 - Much of the IPv6 support is done
- CFURL and NSURL support for IPv6 addresses is planned

[http://\[2002::11FF:1234::1\]/myPage.html](http://[2002::11FF:1234::1]/myPage.html)





What's Next?

Vincent Lubet

The List

Strong End System Model support; Maintainable APIs for Kernel Extensions; IPv6 Firewall; Asynchronous name resolution APIs; Easier to setup 6to4; Easier to setup IKE; Opportunistic IPsec encryption; PPP v6; DHCP v6; IPv6 fully supported thorough the system; Zero copy sockets; Non-privileged ping; PPP dial-in; PPPoA; Setting of Ethernet media; Change MAC address; Jumbo frames; EUI-64 MAC address; QoS APIs; Dummynet; More documentation; . . .



Roadmap

100 The Darwin Road Map

Room A1
Mon., 2:00pm

803 Mac OS X Networking Overview

Room A2
Tue., 9:00am

**301 Cocoa: What's New:
NSURLHandle, NSNetServices**

Civic
Tue., 9:00am

804 Client Web Services Frameworks

Room J
Tue., 10:30am



Roadmap

805 Introducing CFNetwork:
Covers CFNetwork in depth

Room C
Tue., 5:00pm

108 Managing Kernel Extensions:
Best practices apply also to NKE

Civic
Wed., 10:30am

**808 Managing I/O:
CFRunLoop and CFStream:**
Also includes CFSocket

Room C
Wed., 2:00pm

809 Advanced Mac OS X Networking:
IPv6, IPsec, NKE, Performance

Room C
Thurs., 9:00am



Roadmap

811 Zero Configuration Networking:
Rendezvous: Includes CFNetServices

Room J
Thurs., 2:00pm

FF005 Feedback Forum: Toolbox
For InternetConfig feedback

Room J1
Thurs., 10:30am

**FF012 Feedback Forum:
Core OS Networking**
General Mac OS X Networking feedback

Room J1
Fri., 2:00pm

FF016 Feedback Forum: Cocoa
For NSURL, NSURLHandle feedback

Room A1
Fri., 5:00pm



Who to Contact

Thomas Weyer

Network and Communications Evangelist

weyer@apple.com

Vincent Lubet

Manager, Core OS Networking Team

vlubet@apple.com



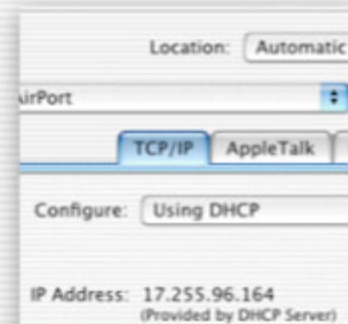
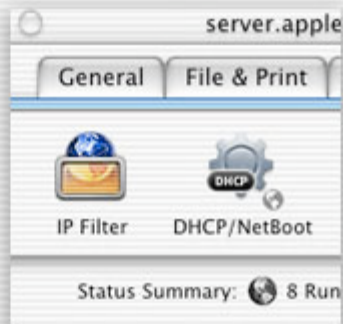
Additional Resources

- Mac OS X
<http://developer.apple.com/macosx>
- Darwin
<http://www.publicsource.apple.com>
- Mac Networking Mailing List
<http://www.lists.apple.com/mailman/listinfo/macnetworkprog>





Q&A



Thomas Weyer
Apple Worldwide Developer Relations
weyer@apple.com

<http://developer.apple.com/wwdc2002/urls.html>

 **WWDC2002**

 **WWDC2002**

 **WWDC2002**